Continue

Pmi acp cheat sheet pdf

How many questions in pmi acp exam. Is pmi acp difficult. Pmi-acp cheat sheet. Pmi-acp study material free download. Pmi acp vs safe.

Agile methodologies prioritize collaboration, shared values, and flexibility. They involve rolling wave planning, iterative delivery, rapid response to change, and open communication among teams, stakeholders, and customers. Key aspects of Agile include: * **Values:** Individuals and interactions over processes and tools * **Principles:** * Customer satisfaction as the highest priority * Embracing change * Frequent delivery * Collocated team * Motivated individuals * Face-to-face communication * Working software * Constant pace * Continuous attention * Simplicity * Self-organization * Regular reflection The Scrum framework also emphasizes certain values: * Commitment * Courage * Focus * Openness * Respect Additionally, the three pillars of Scrum are: 1. Transparency: Clear visibility into work and progress. 2. Inspection: Regular evaluation and adaptation. 3. Adaptation: Continuous improvement. In Scrum, the Product Owner is responsible for maximizing value and managing the backlog. They represent customers, users, and stakeholders. The Development Team consists of self-organizing individuals with diverse skills, typically no more than nine members (excluding the Product Owner and Scrum Master). The Scrum Master plays a crucial role in ensuring the team understands and practices Agile principles. Extreme Programming (XP) values simplicity, communication, feedback, courage, and respect. **Practices** - Pair programming: One coder works on code while another provides navigation. - Test-driven development: Develops tests before writing code. - Continuous integration: Integrates changes frequently to ensure quality. - Refactoring: Improves existing code without changing functionality. - Small releases: Releases software in smaller, manageable chunks. - Sustainable pace: Maintains a steady pace of progress without overloading the team. **Collective Code Ownership** Multiple developers own the codebase collectively. Any pair can improve or amend any part of it. In agile project management, understanding the system's dynamics is crucial. By analyzing the work queue's size and applying Little's Law, teams can improve cycle time by reducing WIP and increasing ACR. Throughput measures the number of features developed in a given timeframe. In Kanban systems, limiting WIP helps alleviate workload pressures, allowing for more creative problem-solving. David Anderson's 5'S for Kanban development emphasizes setting in order, sorting, shining, standardizing, and sustaining processes. Scrumban combines elements of Scrum and Kanban, providing an agile framework with a charter that answers the W5H questions (who? what? where? when? how?). The planning process involves four phases: envisioning, speculating, exploring, and adapting. Agile project management also considers revenue streams, categorizing them into new revenue, incremental revenue, retained revenue, and operational efficiencies. To plan effectively, teams employ various techniques, including design thinking's "Design the Box" and "Prune the Tree" for requirements gathering. Prioritization is essential, using MoSCoW criteria to determine Must-Haves, Should-Haves, Could-Haves, and Won't-Haves. Kano Analysis helps identify Exciter/Delighters, Performance/Satisfiers/Threshold/Must-Have features that deliver expected or unexpected benefits to customers. Ensuring Customer Satisfaction - Neutral vs. Indifferent Features: Prioritizing Delighters and Satisfiers, Minimizing Dissatisfiers. To boost customer satisfaction, focus on delighters and satisfiers while reducing or eliminating indifferent features. Estimation Techniques: 1. **Affinity Estimating**: Group similar user stories together based on common sizes (e.g., T-Shirt sizing). 2. **Wideband Delphi**: Plot anonymous estimates on a chart to minimize biases and reach consensus. 3. **Fist-to-Five, Thumbs Up/Down**, and **Decision Spectrum**: Quickly gather opinions through visual voting methods. Collaboration Games: 1. **Buy a Story**: Understand complex issues by simulating stakeholder decisions. 2. **Brainstorming**: Utilize round-robin, quiet writing, or free-for-all approaches to generate ideas. 3. **Listening Game** and **Collaborative Origami**: Foster active listening and collaboration. Backlog Management: 1. **User Stories**: Support analysis with independent, negotiable, valuable, estimable, specific, and timely stories. 2. **Epic Stories**: Large, unbroken-down stories typically found at the product backlog's bottom. 3. **Story Mapping**: Identify missing stories, categorize by functionality, and prioritize. Spikes: 1. **Architectural Spike** (proof of concept): Validate technical feasibility. 2. **Risk-Based Spike**: Assess potential risks before investing time. Acceptance Criteria: Originates from customers, even if team members write them down. Theming: Group related user stories together as a single unit for easier management. Estimating and release planning rely on measuring scope through user stories, requirements, or features. Risk management is done via the risk-adjusted backlog, prioritized by expected monetary value (EMV). Grooming the backlog involves refinement, adding detail, removing stories, and prioritizing. Release planning includes slicing stories, estimating velocity, and selecting priorities. Milestones are intermediate timeframes, with story-level capability and commitment. The definition of ready determines when an item is development-ready. Staging defines and prioritizes non-functional requirements. Iteration planning breaks down stories into tasks, which team members self-assign or collectively decide upon. Tasks are estimated during iteration planning and re-estimated during the iteration. The product owner shouldn't attend stand-ups, a meeting for the team by the team. The agile project schedule is created by estimating story points and applying velocity. Sprint 0 focuses on initial training. Actual time is the time needed to complete an assignment, while ideal time assumes no interruptions or distractions. To maintain feature buffers, teams can use a MoSCoW scheme to logically sequence work. Avoiding bottlenecks requires generalist team members who can pick up any task. Incremental delivery reduces rework by identifying issues early. Velocity measures the number of story points completed in an iteration and isn't comparable across teams. Teams should strive for predictable velocity, calculating it through team consensus or capacity calculations. Lead time is the time needed to get a feature from inception to deployment, not velocity. Feeding buffers ensure dependencies are met if they're late. Retrospectives follow a specific format. The stage for problem-solving involves setting up a circle or semi-circle, using techniques like Check In, Focus On/Off, ESVP, and Working Agreements. The goal is to generate insights through brainstorming, pair interviews, and force field analysis, deciding what to do using short subjects, triple nickels, and voting with dots. For Agile modeling, use cases, data diagrams, and screen designs are employed. In DevOps, continuous integration (CI) is executed daily when code changes are checked in and tested, utilizing components like source code control systems, build tools, test tools, schedulers/triggers, and notifications. Agile contracts involve graduated fixed price contracts for schedule variance sharing. DSDM contracts focus on work being "fit for business purpose." The "money for nothing" concept, introduced by Jeff Sutherland, allows customers to terminate early if needed. Information radiators include burn charts, retrospective learnings, story maps (excluding definition of done), and sometimes vision statements or roadmaps. Burnup charts provide a bird's-eye view of the project, displaying progress and predicting completion dates. Burndown charts track scope and schedule progress, while cumulative story point burndowns show the total number of story points completed through each iteration. * The bottom of the bar shifts down by 20 points after 50 points are completed, indicating progress. * As the team completes more work, the top of the bar moves down, tracking KPIs such as remaining work rate and likely completion date. * Using ROI (Return on Investment) is a better metric than velocity when comparing Agile teams. * A trend analysis graph helps forecast issues based on trends, making it a leading indicator. * The EMV chart is preferred over GANTT charts due to its ability to track variables over time. * In quality management, escaped defects are those found by customers and not during testing. * Component tests verify that units work together, while error-feedback ratio measures the number of new defects generated when fixing existing ones. * A cumulative defect removal rate of 95% is a nodal point where benefits such as improved quality accrue. * Net Present Value (NPV) and Internal Rate of Return (IRR) are used to evaluate project profitability. * NPV compares projects by calculating the present value of cash flows, while IRR removes assumptions on interest and inflation rates. * The payback period is a calculation where shorter periods are preferred over longer ones. * Earned Value Management (EVM) should be measured at the iteration level in Agile projects. * Kaizen involves making small changes, and the Plan-Do-Check-Act (PDCA) cycle helps implement these improvements. * Common causes are often mistaken for special causes; if it's a common cause, no action is needed. * When learning Agile, teams should "think" first by internally learning principles before implementing changes. You don't have to wait for the end of an iteration. Agile coaching styles include training/teaching, coaching, and mentoring/advising (not supporting). When a team is storming, you should coach with a focus on conflict resolution. There are three levels to improve a team: process, quality and performance, and team dynamics. Shu Ha Ri guides improvement by following the rule, breaking the rule, and being the rule. For one-on-one coaching, Lyssa Adkins' guidelines ensure safety, meet them halfway, partner with managers, and create positive regard. Team building and communication involve servant leadership, shielding the team from impediments, and carrying food and water. Caves and common areas are essential for a team space to allow for quiet work and knowledge sharing. Decision framing focuses on who gets involved in the decision process, not just who makes the decision. Health checks, like questionnaires, help determine adherence to Agile processes. When facilitating meetings, the facilitator is responsible for goals, rules, timing, and assisting, including meeting minutes. During problem-solving, ask questions, listen, but avoid injecting your own ideas. Understand one's feelings, manage them, become more aware of others, and finally influence their feelings. The 5 dysfunctions of a team are absence of trust, fear of conflict, inattention to results, avoidance of accountability, and lack of commitment. Level-based conflict is best solved by empowering the team members themselves (e.g., level 2 conflicts), unless the temperature is high. For announcing sprint priorities, use a two-way communication model for bidirectional information. MISCFDD methodology models Crystal methodology has more ceremony based on the size of the team and criticality of the project. Normative methodologies are based on solutions or sequences of steps known to work for the discipline. Workshops are meetings where work gets accomplished (different from status meetings or knowledge transfer). Learn Cockburn's failure modes and 7 anti-patterns for a methodology. Blanchard and Hersey's situational leadership stages include Directing, Coaching, Supporting, and Delegating. Be aware of the Dreyfus model of adult skill acquisition: Novice, Advanced Beginner, Competent, Proficient, and Expert. Also, be aware of Tuchman team formation and development and Shu-Ha-Ri of skill mastery. Published on March 31, 2016.